



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **03152624 A**(43) Date of publication of application: **28 . 06 . 91**

(51) Int. Cl.

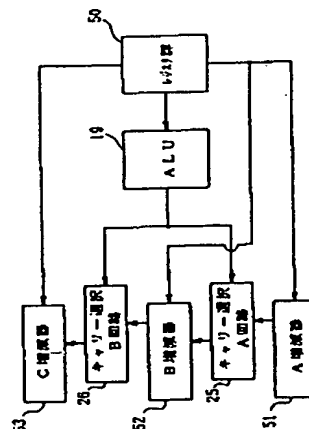
**G06F 7/50**  
**G06F 9/34**
(21) Application number: **01291799**(22) Date of filing: **09 . 11 . 89**(71) Applicant: **RICOH CO LTD**
(72) Inventor: **YASUI TAKASHI**  
**YOSHIOKA KEIICHI**  
**YAMAURA SHINICHI**
**(54) CENTRAL ARITHMETIC PROCESSOR****(57) Abstract:**

**PURPOSE:** To shorten the processing cycle of a central arithmetic processor without losing the parallel arithmetic performance by adding a carry signal produced from an arithmetic process of a data arithmetic unit to the number of adders to be added via an increment device in order to obtain the address data.

**CONSTITUTION:** In the register/indirect 8-bit displacement, an arithmetic and logic unit ALU 19 calculates the 8-bit displacement and the lower 8 bits of the data on a register W2. Then the carry signal obtained from the calculation is selected by a carry selection circuit 25. The circuit 25 sends the carry signal to a B increment/decrement device 52. On the other hand, the high-order data, etc., are transferred to the device 52 and a C increment/decrement device 53. Then the A-C increment devices 51-53 work after deciding the increment or the decrement of a constant 0 based on the code extension value. In a register indirect 16-bit displacement, the arithmetic result is supplied to a carry selection B circuit 26 and then sent to the device 53. Thus the addresses are calculated by the ALU 19 and the devices 51-53. Then the processing cycle is

shortened while maintaining the parallel arithmetic performance.

COPYRIGHT: (C)1991,JPO&amp;Japio



⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A) 平3-152624

⑮ Int. Cl.<sup>5</sup>

識別記号

庁内整理番号

⑬ 公開 平成3年(1991)6月28日

G 06 F 7/50

E  
Z

7056-5B

7056-5B

9/34

7927-5B

G 06 F 9/36

3 1 0

審査請求 未請求 請求項の数 1 (全12頁)

⑭ 発明の名称 中央演算処理装置

⑰ 特 願 平1-291799

⑱ 出 願 平1(1989)11月9日

⑲ 発 明 者	安 井	隆	東京都大田区中馬込1丁目3番6号	株式会社リコー内
⑲ 発 明 者	吉 岡	圭 一	東京都大田区中馬込1丁目3番6号	株式会社リコー内
⑲ 発 明 者	山 浦	慎 一	東京都大田区中馬込1丁目3番6号	株式会社リコー内
⑳ 出 願 人	株 式 会 社	リ コ ー	東京都大田区中馬込1丁目3番6号	
㉑ 代 理 人	弁 理 士	青 山 葆	外1名	

明 細 書

1. 発明の名称

中央演算処理装置

2. 特許請求の範囲

(1) アドレスのビット数を複数に分割したビット数を処理し、供給される複数の定数のいずれか一つを被増加数に加算しアドレスデータを作成する複数の増加器と、

少なくとも一つのレジスタと、

上記レジスタが送出する、アドレスのビット数より少ないビット数からなるデータに基づき演算を行う一つのALUと、

上記ALUの演算結果で桁上げ信号が発生した場合にはこの桁上げ信号を上記増加器に送出するキャリー信号選択手段と、を備えたことを特徴とする中央演算処理装置。

3. 発明の詳細な説明

[産業上の利用分野]

本発明は、中央演算処理装置に関する。

[従来の技術とその課題]

中央演算処理装置においてデータの演算処理を実行する際に必要なアドレス信号を発生する方法として、データ演算用のALUとは独立して設けられるフルアダーにて構成されアドレス信号のみを発生するアドレス計算ユニットを設ける方法と、データの演算とアドレス信号の発生との両方をもに行う一つのALUを設ける方法とがある。

前者のアドレス計算ユニットを設ける方法においては、ALUとアドレス計算ユニットとを別々に設けなければならない、ハードウェアが大きくなるという問題点がある。一方、後者の一つのALUを設ける方法においては、データ演算とアドレス計算とを行うためそれぞれの演算を別々に行うことができず並列性がなくなり、演算に要する処理サイクルが多くなるという問題点がある。

本発明はこのような問題点を解決するためになされたもので、ハードウェアが大きくなり、かつ演算の並列性を損なわない中央演算処理装置を提供することを目的とする。

[課題を解決するための手段]

本発明は、アドレスのビット数を複数に分割したビット数を処理し、供給される複数の定数のいずれか一つを被増加数に加算しアドレスデータを作成する複数の増加器と、

少なくとも一つのレジスタと、

上記レジスタが送出する、アドレスのビット数より少ないビット数からなるデータに基づき演算を行う一つのALUと、

上記ALUの演算結果で桁上げ信号が発生した場合にはこの桁上げ信号を上記増加器に送出するキャリー信号選択手段と、を備えたことを特徴とする。

#### 【作用】

キャリー信号選択手段は、ALUにおける演算処理にてキャリー信号が発生した場合にはこのキャリー信号を増加器に送出し、増加器は被増加数にキャリーデータを加算しアドレスデータを発生する。このようにALUと増加器とによりアドレスデータを発生することは、ALU単体でアドレスデータを発生する場合に比べ並列性を失わず処理

を用いており、バンクアドレスとしては、基本的にデータバンクレジスタ(以下DBRと記す(8ビット))が出力され、従って、64Kバイトリニアで256バンクを用いて、16Mバイトのアクセスを可能にしている。

尚、バンクアドレスとしてのDBRの出力については後述のM1,M0フラグにて説明する。

また、複数の汎用レジスタ(W0~W3:16ビット)があり、特に、W0,W1の両レジスタは8ビットごとに分割され、8ビットレジスタR0,R1,R2,R3として使用することもできる。

故に、本CPUでは、演算のデータサイズとして8ビット、16ビットの両方のサイズのデータを命令により区別して扱うことが可能である。

さらに、スタック空間としては、スタックポインタレジスタ(以下SPと記す)として16ビットレジスタを用意しており、リニアに64Kバイトのアクセスをおこなう。ただし、バンクアドレスは、"00'hに固定されている。

そして、プログラムステイタスレジスタ(以下

サイクルを短くするように作用し、又、アドレスデータ計算用のALUを有する必要がないことによりハードウェアが小さくなるように作用している。

#### 【実施例】

まず、本発明の中央演算処理装置における一実施例における構成の概略を第3図ないし第5図a,b,cを参照し以下に説明する。

第4図は、プログラミングモデルであり、本中央演算処理装置(以下CPUと称す)の基本語長は8ビットである。

アドレス空間は、プログラムをアクセスする際、プログラムカウンタ(以下PCと記す)は24ビット(PBC、PCH、PCL)を有しており、リニアアドレスで16Mバイトをアクセス可能としている。尚、PBCとはプログラム・バンク・カウンタレジスタ(以下PBCと記す)、PCHとはプログラム・カウンタレジスタH(以下PCHと記す)、PCLとはプログラム・カウンタレジスタL(以下PCLと記す)である。

一方、データをアクセスする時は、バンク方式

(PSRと記す)は現在のCPUの動作状態を示しており、具体的には、N,V,Z,Cの各フラグは、演算の結果により変化し、Iフラグは、割り込み要求の受付の可否を示し、Dフラグは、加減算命令の結果の補正に関し、D=1ならば、加減算命令の実行結果は自動的に10進補正される。

M1,M0フラグは、データ空間をアクセスする際、出力されるバンクアドレスの選択を可能にするフラグである。従って、M1,M0フラグを任意の値に設定(このCPUでは命令で更新する)する事により、データアクセスの際に、出力されるバンクアドレスをDBR値、"00'h等の定数の何れかを選択して出力し様々なメモリのアプリケーションに対応させる。

ファーストページレジスタ(以下FPRと記す)はデータアクセス時のアドレス・ポイントとなるレジスタで、ファースト・ダイレクトと呼ぶアドレス・モードで使用される。尚、アドレッシングとは、データの格納先のアドレスを指定することをいう。

このアドレッシング・モードではオペランド・データとして8ビットのデータのみをフェッチしてそのデータを実効アドレスのロー(ビット7～ビット0)とし、ハイ(ビット15～ビット8)をFPRの内容とするアドレッシングモードにおいて有効となるレジスタである。

ただしこのときも、出力されるバンクアドレスは、M1、M0のフラグ状態に従う。

このアドレッシング・モードは、オペランドデータを1バイトのみフェッチするだけなので、同一ページアドレス内(アドレスのビット15～ビット8が一定値)の高速なデータのアクセスが可能となる。

第5図aないし第5図cは、本CPUの命令形式について示したものであり、このCPUは基本語長は、前述のように8ビットであり、オペコードの前にプリバイトと呼ばれる命令拡張用の1バイトデータをフェッチする形式をとる。

基本的にプリバイト・データは、アドレッシング・モードに係る情報を有し、オペコードが実行

することで実効アドレスを発生するアドレッシングのことを示す。

このアドレッシング・モードが使用される際、もし形式2のようなオペランドデータの配置形式を取れば、ディスプレースメントのオペランドデータをフェッチした後、実効アドレスを計算するために、時間を要し、オペランドのディスプレースメント・データのフェッチの後、複数のアイドルサイクルが存在することになる。

しかし、形式3の配置をこのとき用いて、プリバイトとオペコードの間にディスプレースメントデータを配置すれば、実効アドレス発生のための計算をオペコードのフェッチサイクルに重複して行なう事ができ、無駄なアイドルサイクルの発生を防ぐ。

第3図は、ブロックレベルの構成図であり、本CPUは主に制御部1と演算部2の2つの機能部に大別される。

初めに、制御部1であるがここは、命令の実行を制御する機能を持っている。

すべき命令の内容を持っている。

但し、命令の使用頻度が高いものについては、命令コード長と実行時間の短縮を図るため、「形式1」に示すように、短縮命令と呼ぶプリバイトの無いオペコード内にアドレッシング及び命令の内容を含んだ命令を用意する。

さらに、オペランドデータは2種類の配置形式をもっている。第5図bに示す「形式2」は、プリバイトの次にオペコードを配置し、その後オペランドデータを配する形式であり、「形式3」はプリバイトとオペコードの間にもオペランドデータを配置する。

特に、形式3のプリバイトとオペコードの間のオペランドデータは、ディスプレースメント付アドレッシングで使用される。

ここでいうディスプレースメント付アドレッシングとは、データのアクセスのための実効アドレスの発生時に、内部レジスタデータにオペランドでフェッチされたデータもしくは、オペランドで指定されたレジスタの値をオフセットとして加算

動作としては、命令の実行に際し、外部からデータバス(D7～D0)を介して、D11～D15に入力された命令コードは、プリバイトIR3或いはオペコードIR4の各インストラクションレジスタに格納され次の命令が発生するまで保持される。

そして、これらのインストラクションレジスタの複数の出力5、6と命令シーケンスのタイミングを制御するTCU7の出力がAND-ORのPLAで構成された命令デコード回路8、9、10、11に入力され、命令とタイミングに応じたデコード結果13を出力する。

さらにそのデコード結果は、EC112というインターフェース回路を介して、演算部2に対してタイミングを整えて演算部2を制御すべき複数の制御信号14を発生する。

但し、本CPUにおいて、PLAの構成は、ANDプレーンをプリバイト用(構成部分8)と、オペコード用(構成部分10)の2種類もち、ORプレーン9、11を共有した形をとる。

これは、先の命令形式でも記述した様に、プリ

バイト部は、アドレッシングモードの情報を有し、オペコード部が命令のオペレーション内容を含むため、PLA上でも機能的に、分類することでコードの容易化と冗長性を排除し、機能別(プライベートかオペコード)で最小のPLA(特に、ANDプレーン)を実現させている。

そして、この2分割されたPLAのANDプレーン8,10は、インタラプト制御21からの入力信号24により、ANDプレーンの両方を動作状態にするか、一方ANDプレーン10を非動作状態にせしめることもできる。ここで割り込みのシーケンスの制御コードは、全て、プライベート側のANDプレーン8にコードが割り付けられており、割り込みの処理時にオペコード側のANDプレーン10は非動作状態にある。

演算部2は、上記の制御信号にしたがって、演算やCPU外部とのデータのアクセスを行なう。

内部バスとしては、基本的にMB,DB,SBの3種類8ビットバスを有し、各機能部とのデータのやりとりを行なう。

具体的にはACU部は、8ビット毎にINC/DECという増減機能があり、ABL,ABH,SBという内部バス(各8ビット)からのデータを"00'h,"01'h,"02'hで選択的に増減する。

INC/DECで演算された結果は、CALL,CALH,CALBのラッチに選択的に格納され、AOBL,AOBH,AOBBのアドレス・バッファを介して出力される。

ここで選択的というのは、演算結果が常にラッチされるのではなく、アドレス演算時のみラッチして、データ演算時にはラッチされない場合があることを意味する。

しかし、RLT2 35は、INC/DEC:B演算時は常に結果をラッチするデータラッチである。

ACU部には、割り込み発生時に強制的に割り込みベクタを発生するVECL,VECH,VBCB(ベクタアドレス発生回路)や、INC/DECを介さずにDBバスデータを直接アドレスとして出力するBSも配置されている。

機能としては、上述のプログラミングモデルで示したレジスタ群や、データや実効アドレスの演算を行なう8ビットALU19や、シフト演算を行なう8ビットのシフト20、アドレス生成を主に行なうACU13がある。

ALU19は、MB入力側にIC27をもち、IC27は、MBバスから入力される信号を、スルーするか、反転したり、"00'h等の定数データを発生してALU19での演算を補助する。

さらにDフラグの機能を実現するための10進補正回路もALU19は含んでいる。

そして内部バス(MB)のデータのゼロを検出するZDT17や分岐命令での分岐条件成立の有無をPSRの状態から検出するBRDT18もある。

特にアドレス生成を主に行なうACU部に関しては、8ビット単位に、機能が分離されそれぞれはキャリーが伝搬する構成となっていて、最大24ビットのアドレス演算を行なう。ここでは、アドレスの演算のみならずデータの演算も可能である。

本CPUにおいて、実効アドレスの生成は、特に分岐やディスプレースメント付のアドレッシングにおいてAUとACUの両方を使用して演算しており、CSB,CSH25,26は、その際に使用される。

つまり、ALU19からの演算結果によるキャリーやボローをACUの演算に反映させるためのキャリーのセレクトとしての機能をCSB,CSH25,26が持っている。

尚、INC/DECからラッチされた演算結果は、SB,ABH,ABLのバスを介してPC,DBR,TR,ADH,ADLのレジスタデータを選択的に更新する。

その他の機能としては、CPUのクロックの制御をつかさどる、クロック発生器22や、周辺システムにCPUの動作状態を知らせる複数の信号を発生するシステム制御23がある。

さらに、インストラクション・プレデコード33は、命令コードのプリデコードを行い短絡命令の識別や、プライベート付でオペコードと不当な組

合せ(以下不当命令と称す)の選別などを行なう。

以下に、本CPUの演算部2の各機能部について説明を行なう。

#### ○ 汎用レジスタ

演算、転送時にデータを提供したり、演算、転送後の結果を格納する第3図及び第4図に示す汎用レジスタ群である。

W0、W1については、8ビットずつに分けてR0、R2、R1、R3の8ビットレジスタとしても命令で区別して使用することができるので、本CPUでは16ビットのみならず、8ビットのデータを扱うことができる。

W2、W3は、データアクセスの際のポインタとしてアドレッシングモードで指定すれば使用することもできる。

汎用レジスタ群の各レジスタは、ラッチ(セット、リセットなし)で構成され、内部バスに対し、以下の接続関係を有する。

基本的に、MBバスから入力されデータをラッチし、DBあるいはMBのバスにラッチされたデータ

基本的に、MBバスから入力されデータをラッチし、DBのバスにラッチされたデータを、出力する。

FPR → MBから入力、DBへ出力

#### ○ IC(ALUに関する入力制御)

第3図に示したIC27(8ビット)は、MBバスからALU19に入力されるデータを制御する。機能的には、以下の機能を有する。

1. MBバスデータ → ALUに入力
2. MBバスデータの反転 → ALUに入力
3. "00"hの定数 → ALUに入力  
(MBバスデータは無視する。)
4. "01"hの定数 → ALUに入力  
(MBバスデータは無視する。)
5. "02"hの定数 → ALUに入力  
(MBバスデータは無視する。)
6. "03"hの定数 → ALUに入力  
(MBバスデータは無視する。)

#### ○ ALU(演算論理素子)

第3図に示したALU19(8ビット)は、DB

タを出力する。

R2レジスタのみ、除算命令を実行する際のため、入力にDBバスを選択可能とする。

R0(W0L) → MBから入力、DBあるいはMBへ出力

R2(W0H) → MBあるいはDBから入力、DBあるいはMBへ出力

R1(W1L) → MBから入力、DBあるいはMBへ出力

R3(W1H) → MBから入力、DBあるいはMBへ出力

W2L → MBから入力、DBあるいはMBへ出力

W2H → MBから入力、DBあるいはMBへ出力

W3L → MBから入力、DBあるいはMBへ出力

W3H → MBから入力、DBあるいはMBへ出力

#### ○ FPR(ファーストページレジスタ)

第3図及び第4図に示したFPRは、前述のファースト・ダイレクト・アドレッシングと呼ぶアドレッシングモードで使用される。

FPRは、ラッチ(セット、リセットなし)で構成され、内部バスに対し、以下の接続関係を有する。

バスデータとICからの各8ビットの入力により演算を実施する。

機能的には、AND(論理積)、OR(論理和)、EXOR(排他的論理和)、SUM(加算)がある。

また、PSR中のDフラグの設定により(D=1ならば)、加算及び減算を同一演算サイクル内で10進補正する回路も含む。

さらに、SUMの結果、キャリー・ポロー発生、オーバーフローが発生の検出及びキャリー・ポロー、オーバーフローをラッチする機能も具備する。

特に、キャリー結果は、ALU19が次のSUMを実行するまで保持されるものとする。(AND、OR、EXORでは変化しない)

#### ○ ALUシフト(演算論理素子シフト)

第3図に示した、ALUシフト28は、8ビットデータの1ビットシフトライトを実施するシフトレジスタで、主に乗算命令で使用される。

このシフトレジスタに入力されるデータは、ALU19のSUM(加算)の結果であり、最上位ビットには、そのSUMの結果で発生したキャリーが

入力され、シフトの結果最下位より送出される1ビットデータは、ALU19のキャリーとして最終的に保持される。

○ RLT(ALU 結果ラッチ)

第3図に示した、RLT29は、ALU19の演算結果を保持する8ビットラッチである。内部バスに対しては以下の接続関係を有する。

RLT → DBあるいはMBへ出力

ただし、RLT29のデータは、次のALU演算が実行されるまで内容は更新されない。

○ シフト

第3図に示した、シフト20はフリップ・フロップで構成され、データの1ビットシフトレフト、シフトライト、ノーシフトのいずれかを制御信号により選択的に実施する。

内部バスに対しては以下の接続関係を有する。

シフト → MBから入力、MBへ出力

○ ZDT(ゼロ検出回路)

第3図に示したZDT17は、MBバスの状態をモニタし、MBバスが全ビット"00"hならば、

"00"hの検出をしたことを示す信号を発生するゼロ検出回路である。

特に、この信号はPSRレジスタ30中のZフラグに作用し、ALU19等の演算結果がRLT29より、MBバスに出力される時、結果の"00"hを検出してZフラグを"1"にセットする動作を促すために用いられる。

○ PSR(プロセッサ・ステータス・レジスタ)

第3図に示した、PSR30は、ラッチで構成され内部バスに対しては以下の接続関係を有する。

PSR → MBから入力、DBへ出力

機能としては、概要でも記述した様にPSRレジスタ30は現在のCPUの動作状態を示す。

○ BRDT(分岐検出回路)

第3図に示した、BRDT18は、PSR30に接続されており、分岐命令が発生した場合、PSR30の内容から分岐するか否かを判断する信号を発生する。

○ AOBH,AOBH,AOBL(アドレス・出力バッファ)

である。

○ INC/DEC:B:H:L(インクリメント/デクリメント・ユニット)

第3図ACU部16に示したINC/DEC:B:H:Lはデータの増減を行なう。

各機能部は、8ビット単位で構成され、演算結果で発生したキャリーは、それぞれの上位アドレス増減部(INC/DEC:LならINC/DEC:Hへ、INC/DEC:HならINC/DEC:Bへ)に伝搬され、結局24ビットのアドレス生成を実現することになる。

但し、このINC/DEC:B:H:Lにデータ(各8ビット)は、SB,ABH,ABLのデータバス(各8ビット)を介して入力される。

各INC/DEC:B:H:Lは、このデータについて基本的に次の動作を選択的に行なう。

1. 現状データの保持
2. "01"hのインクリメントあるいはデクリメント。
3. "02"hのインクリメントあるいはデクリ

第3図ACU部16に示した、AOBB,AOBH,AOBLはアドレス出力用のバッファであり、各8ビットで計24ビット(BA7~BA0,A15~A0)のアドレスを出力する。

アドレス出力は、BEのローでハイ・インビデンス状態になる。

○ VECB,VECH,VECL(ベクタアドレス発生器)

第3図ACU部16に示した、VECB,VECH,VECLは割り込み処理において、ベクタアドレス(24ビット)を発生する。

○ CALB,CALH,CALL(アドレス計算ラッチ)

第3図ACU部16に示した、CALB,CALH,CALLはINC/DEC:B:H:Lの演算の結果を選択的に格納するラッチであり、アドレス演算時のみラッチされる。

○ RLT2(結果ラッチ2)

第3図ACU部16に示した、RLT2はINC/DEC:Bの演算の結果を常に格納するラ

メント。

○ BS(バス セレクト)

第3図ACU部16に示した、BSは実効アドレスを発生する際、CPU外部から入力されたデータ(8ビット)をINC/DEC:Lを介することなく、DBバスから、直接AOBLに入力するためのデータの選択の機能を有する。

前記のファースト・ダイレクト・アドレッシングのような場合、実効アドレスのためのオペランドデータ(8ビット)をフェッチするサイクルの次に、すぐに実効アドレスを出力しなければならないが、この場合、INC/DEC:Lを介すれば遅延が生じる。

そこでこのBSを用いて、オペランドデータ(DIL)をDBバスに乗せ、BSで選択することにより、高速にAOBLを書き換えることができる。

○ CSB,CSH(キャリーセクタ)

第3図ACU部16に示した、CSB,CSH25、26はデータの演算時に、INC/DEC:B,INC/DEC:Hに入力されるキャリーが

めて演算ができる。

一方、通常のプログラムカウンタのインクリメント動作の場合には、ACU16のみを用いて、ALU19は別のオペレーションのための演算を行なうことができる。

この時ALU19のキャリーは無視されACU16から発生したキャリーがCSH25を介してACUHに入力される。

○ PBC,PCH,PCL(プログラム・カウンタ)

24ビットのプログラム・カウンタ・レジスタである。

このレジスタのインクリメントは、INC/DEC:B:H:Lを用いて行なう。

内部バスに対しては以下の接続関係を有する。

PBD → SBから入力、DBあるいはSBへ出力

PCH → ABHから入力、MBあるいはABHへ出力

PCL → ABLから入力、DBあるいはABLへ出力

○ TR,ADH,ADL(テンポラリ・レジスタ)

INC/DECの下位側(INC/DEC:HならINC/DEC:L,INC/DEC:BならINC/DEC:H)からか、あるいはALU19で発生されたキャリーにするかを選択する機能を有する。

従って、このCPUでは実効アドレス発生の際のディスプレイメントデータの加算や、プログラム相対アドレスで分岐の際にアドレスの計算を行なうことは、ALU19とACU16を共用して行なう。

例えば、24ビットデータに8ビットのディスプレイメントを加算して、実効アドレスを発生するアドレッシングの場合、24ビットデータ中のビット7～ビット0とディスプレイメントデータ(8ビット)の加算をALU19で行ない、24ビットの残り(ビット23～ビット16)をACU部16で演算する。

ALU19で加算の結果キャリーが発生した場合、このキャリーは、CSH25を介して、ACUHに入力され、ACU16はこの桁上りを含

各8ビットのテンポラリ・データラッチである。CPU外部からは見えない。演算結果を一時的に格納する。

TR → DBあるいはSBから入力、SBへ出力

ADH → MBあるいはABHから入力、ABHへ出力

ADL → DBあるいはABLから入力、ABLへ出力

○ SPH,SPL(スタック・ポインタ・レジスタ)

16ビットのスタック・ポインタ・レジスタである。

内部バスに対しては以下の接続関係を有する。

SPH → MBから入力、MBあるいはDBへ出力

SPL → MBから入力、DBへ出力

○ DBR(データ・バンク・レジスタ)

8ビットのバンク・レジスタである。基本的にデータアクセスの際のバンクアドレスは、このレジスタ値が出力される。但し、PSR中にモード・フラグ(M1,M0)の状態により、バンクアドレス値は変動する。

また、DBRは、SBバスを介して入力されて

おり、DBR値の増減にも任意に対応できる。

内部バスに対しては以下の接続関係を有する。

DBR → MBあるいはSBから入力、DBあるいはSBへ出力

○ DIL(データ入力ラッチ)

8ビットのラッチである。外部データは、このラッチに入力される。

DIL15は、制御部1に対しては命令コードを供給し、演算部2には、内部バス(DB,MB,SB)に対しデータを供給する。

CPU内部に対しては以下の接続関係を有する。

DIL → D7~D0から入力、DB,MB,SBあるいは制御部へ出力

○ DOL(データ出力ラッチ)

8ビットのラッチである。外部に出力されるデータは、このラッチに入力される。

CPU内部に対しては以下の接続関係を有する。

DIL → DBあるいはMBから入力、D7~D0へ出力

以下に、本CPUの制御部1の各機能部について説明を行なう。

○ システム制御

CPUの動作状態を知らせるための複数の信号を発生する。

BSVT --- プロセッサ動作状態出力  
(ベクタアドレス出力中を示す)

BSDA --- プロセッサ動作状態出力  
(データアクセスを示す)

BSPA --- プロセッサ動作状態出力  
(プログラムアクセスを示す)

BSOF --- プログラム動作状態出力  
(命令フェッチを示す)

BSML --- プロセッサ動作状態出力  
(メモリロック状態を示す)

RWB,RB,WB --- リードライト状態出力

BE --- バイネーブル入力

○ インタラプト制御

CPUの割り込みを制御する。

RES --- リセット割り込み入力

NMI --- ノンマスカブル割り込み入力

IRO --- 割り込み入力

○ インストラクション・プレ・デコード

基本的に次の3つの機能部を有する。

1. PLAでのデコードではタイミング的に間に合わない場合、プレデコードで予めデコードして制御信号を発生する。

[1サイクル命令の検出,外部制御信号の発生制御,TCU7の制御等]

2. PLAコードの最小化のためデコードを補助する。

[短縮命令の検出,命令で扱うデータサイズの検出等]

3. 不当命令,ソフトウェアインタラプト命令の検出。

○ クロック発生器

CPU内部用のクロックの発生。あるいは、外部システム用システム・クロックを発生する。

WAIT --- プロセッサ停止入力

LSP --- バスサイクル変更入力

CLK --- CPUクロック入力

S1,S2 --- システム・クロック出力

ISE0~3 --- 割り込み(IRO)選択入力

WAKE --- プロセッサ停止命令の解除入力

○ TCU(タイミング制御ユニット)

命令実行の動作シーケンスを制御する。

○ ECI(イクスキュージョン制御インタフェース)

PLAの命令デコード結果を受け演算部2にタイミングを整えた制御信号を発生する機能を有す。

○ オペコードIR(パッファ)。

ブリバイトIR(インストラクション・レジスタ)

命令を格納するインストラクションレジスタ。

○ プリバイト ANDブレン,オペコード

ANDブレン,ORブレン

AND-ORで構成された命令デコード用PL

A。

以上概説したようなCPUにおいて、実行アドレスを計算する構成部分のみを第3図より抜き出し第1図に示す。尚、第1図に示す構成部分の内、

第3図に構成部分に相当するものについては同じ符号を付している。

本実施例のCPUは各メモリの実行アドレス計算をデータの演算実行用のALUを用いて、かつ増減器を用いて行うが、実行アドレスを計算する際、実行アドレスを構成する例えば24ビットのすべてを計算する必要はなく、本CPUでは、実行アドレスを生成する加算数と被加算数とのデータを8ビットずつ3分割し、ALUにてそれぞれの下位側よりアドレスを計算し下位側からの桁上がりが発生した場合にはそのキャリーを上位側の増減器に送出し加減算を行い実行アドレスを生成する。

第1図において、レジスタ群50は、第3図に示すFPR、R0、W2L、等の汎用レジスタであり、このレジスタ群50の出力側は、A増減器51、B増減器52、C増減器53に接続されるときともに、ALU19に接続される。

A増減器51の出力側は、ALU19より供給される信号とA増減器51より供給される信号と

とW2あるいはW3レジスタのデータとに、オペランドデータとして8ビットあるいは16ビットのディスプレイメントを符号拡張したアドレスが実行アドレスとなる。

ロングダイレクト・16ビットレジスタディスプレイメントでは、オペランドデータとしてプログラムから読んだ24ビットのデータに、レジスタW2あるいはW3のディスプレイメントデータを符号拡張して加えたアドレスが実行アドレスとなる。

まず、レジスタ・インダイレクト・8ビットディスプレイメントでは、8ビットのディスプレイメントとレジスタW2のデータの下位8ビットをALU19で計算し、その結果におけるキャリー信号がキャリー選択A回路25に送出され、キャリー選択A回路25は、ALU19からのキャリー信号をB増減器52へ送出する。一方、B増減器52及びC増減器53には、それぞれレジスタW2の上位データ、DBRのデータが転送され、A、B、Cの各増減器51、52、53は、符号

のいずれかを選択するキャリー選択A回路25を介してB増減器52に接続される。B増減器52の出力側は、B増減器52より供給される信号あるいはALU19より供給される信号のいずれかを選択するキャリー選択B回路26を介してC増減器53に接続される。

尚、A増減器51は、第3図に示すINC/DEC:Lに相当し、B増減器52はINC/DEC:Hに相当し、C増減器53はINC/DEC:Bに相当する。又、キャリー選択A回路25は第3図に示すCSHにであり、キャリー選択B回路26はCSBである。

このように構成される本実施例のCPUにおける動作を以下に説明する。

本CPUの概略にて説明したように、本CPUが有する3つのアドレッシングのアドレスの発生方法を第2図aないし第2図cに示す。

レジスタ・インダイレクト・8ビットまたは16ビットディスプレイメントでは、レジスタ群50内の複数のレジスタより読み出したDBRデー

拡張の値より定数0の増加か定数0の減少かを選択して動作を実行する。このとき、キャリー選択B回路26は、B増減器52からのキャリー信号を選択する。そして、B増減器52、C増減器53の増減結果とALU19の結果が、計算された実行アドレスとなる。さらに、16ビットデータの場合には、もう1バイトアクセスする必要があるため、上記の計算されたアドレスを増減器で定数1の増加をさせて望むアドレスを得る。

次に、レジスタ・インダイレクト・16ビットディスプレイメントでは、16ビットディスプレイメントの下位8ビットとレジスタW2の下位8ビットをALU19にて計算し、その結果をA増減器51に送出し、さらにそのキャリー信号を使用して16ビットのディスプレイメントの上位8ビットとレジスタW2の上位8ビットをALU19にて計算する。その結果のキャリー信号がキャリー選択B回路26に供給され、キャリー選択B回路26はALU19からのキャリー信号をC増減器53に送出する。一方、C増減器53には、

DBRのデータが転送され、増減器全体は符号拡張の値により定数0の増加か定数0の減少かを選択して動作を実行する。このとき、キャリー選択A回路25は、A増減器51からのキャリー信号を選択する。そしてC増減器53の増減結果、ALU19の結果、A増減器51の増減結果が計算された実行アドレスとなる。

次に、ロングダイレクト・16ビットレジスタ・ディスプレイメントの場合は、オペランドとレジスタの関係がレジスタ・インダイレクト・16ビット・ディスプレイメントと入れ代わるだけであり、動作は同じである。

このようにALU19とA、B、Cの各増減器51、52、53にてアドレスを計算することで、ALU単体で計算する場合に比べて演算の並列性を失わず、演算処理サイクルを短くすることができる。

又、ALU及び増減器で計算することにより、データ演算用のALUの他にアドレス計算専用のフルアダーを有する場合に比べてハードウェアを

小さくすることができる。

#### 【発明の効果】

以上詳述したように本発明によれば、演算処理を行うALUにて演算処理の結果発生するキャリー信号を増加器にて被加算数に加算することでアドレスデータを得るようにしたことより、演算の並列性を失わず、処理サイクルを短くすることができ、又、アドレスデータ計算専用のALUが不要であるからCPUのハードウェアを小さくすることができる。

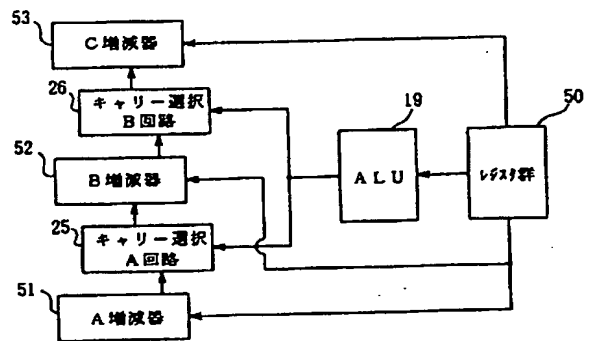
#### 4. 図面の簡単な説明

第1図は本発明のCPUの構成部分の内、アドレス計算にかかる構成部分を示したブロック図、第2図aないし第2図cは本発明のCPUが有するアドレッシングのアドレスの発生方法を示す図、第3図は本発明のCPUの全体構成を示すブロック図、第4図は本発明のCPUのプログラミングモデル、第5図aないし第5図cは本発明のCPUの命令形式について示した図である。

19…ALU、25…キャリー選択A回路、  
26…キャリー選択B回路、51…A増減器、  
52…B増減器、53…C増減器。

特許出願人 株式会社リコー  
代理人 弁理士 青山 藤 外1名

第1図







特開平3-152624

【公報種別】特許法第17条の2の規定による補正の掲載  
【部門区分】第6部門第3区分  
【発行日】平成10年(1998)9月11日

【公開番号】特開平3-152624  
【公開日】平成3年(1991)6月28日  
【年通号数】公開特許公報3-1527  
【出願番号】特願平1-291799  
【国際特許分類第6版】

G06F 9/34  
7/50

【F1】

G06F 9/36 310  
7/50 E  
Z  
9/36 330 B

## 手続補正書

平成 8 年 10 月 25 日

特許庁長官殿

1. 事件の表示

平成01年特許願第2917-99号

2. 補正をする者

事件との関係 特許出願人

名称 株式会社リコー

3. 代理人

住所 〒540  
大阪府大阪市中央区城見1丁目3番7号 IMPビル  
青山特許事務所  
電話(06)949-1261  
FAX (06)949-0361

氏名 弁護士 (0214) 青山 義



4. 補正命令の日付

白紙(出願審査請求と同時に)

5. 補正の対象

明細書:「発明の詳細な説明」の欄

6. 補正の内容

明細書中、次の箇所を補正します。

発明の詳細な説明の欄

- (1) 第14頁第3行に「AU」とあるを「ALU」と補正する。
- (2) 第14頁第18行に「ブレ」とあるを「ブリ」と補正する。
- (3) 第14頁第19行に「行い」とあるを「行ない」と補正する。
- (4) 第21頁第20行から第22頁第1行にかけて「ラッチ」とあるを、「フリップフロップ」と補正する。
- (5) 第25頁第11行、同頁第13行、第26頁第9行、及び同頁第15行に「レジスタ」とあるを「ラッチ」と補正する。
- (6) 第25頁第16行に「PBD」とあるを「PBC」と補正する。
- (7) 第29頁第20行、及び第30頁第1行に「IRO」とあるを「IRQ」と補正する。

以上